**Note**

Tuesday, July 12, 2022

4:38 PM

* + <https://learngitbranching.js.org/?locale=en>
  + <https://learngitbranching.js.org/?locale=vi>

**Task**

Tuesday, July 12, 2022

4:39 PM

**Git overview**

Yêu cầu, đọc và thực hành hết các phần trong này (trừ các phần Optional).

* Git là gì?
* Git làm được gì?
* Các khái niệm cơ bản
  + Repository
  + Object store
  + Index
  + Branch
  + Git work-flow
  + Staging Area

* Git & Github & Gitlab giống/khác gì nhau?
* Các trạng thái có thể có trong git repository: Untracked, Modified, Umodified, Staged. Tham khảo: <https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository>

**Config cho git với git config**

* Config thông tin cá nhân
* Config default/prefered editor
* Config global .gitignore

**Các thao tác ban đầu**

* Khởi tạo repo với git init
* Clone repo với git repo

**Git commit**

* Commit trong git là gì
* Một commit chứa những thông tin gì
* Amending

**Làm việc với branch**

* Tạo mới branch
* Nhảy từ branch này sang branch khác
* Đổi tên branch
* Lab về git branch ở đây: <https://learngitbranching.js.org/>

**Git checkout**

* Check out đến commit ID

**Các thao tác liên quan khác:**

* Git Merge
* Git Rebase
* Git Rebase interactive
* Git Rebase khi remote có cập nhật mới
* Git Cherry-pick
* Pull
* Push
* Tùy chọn force
* Git reset

**Các thao tác với git remote**

* Add remote
* Rename remote
* Update remote
* Prune remote branch

**Các thao tác với Stash**

* Tổng quan về git stash
* git stash list
* git stash clear
* git stash --help

**Git với giao diện đồ họa, Git GUI (optional)**

* gitk
* Git Kraken
* Tham khảo thêm <https://git-scm.com/download/gui/linux>

**Git Overview**

Tuesday, July 12, 2022

2:04 PM

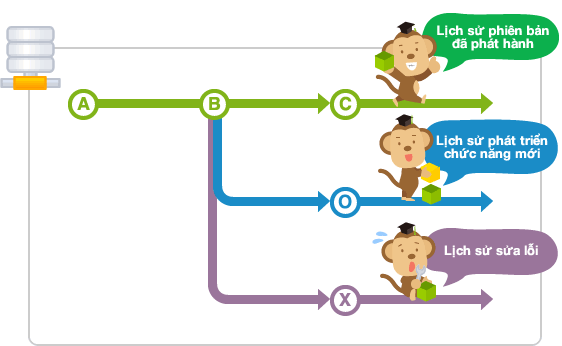
* **Git là gì?**   
  Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS). Là 1 hệ thống giúp cho việc quản lý tài liệu, source code... của 1 nhóm các developer cùng làm chung dự án.
* **Git làm được gì?**
  + Git sẽ ghi nhớ lại toàn bộ lịch sử thay đổi của source code trong dự án. Qua đó giúp dự án có thể điều tra nguyên nhân gây lỗi hệ thống, tổng hợp code trở nên dễ dàng hơn
  + Lưu lại được các phiên bản khác nhau của mã nguồn dự án phần mềm.
  + Khôi phục lại mã nguồn từ một phiên bản bất kỳ
  + Dễ dàng so sánh giữa các phiên bản.
  + Phát hiện được ai đã sửa phần nào làm phát sinh lỗi
  + Khôi phục lại tập tin bị mất
  + Dễ dàng thử nghiệm, mở rộng tính năng của dự án mà không làm ảnh hưởng đến phiên bản chính (master branch)
  + Giúp phối hợp thực hiện dự án trong nhóm một cách hiệu quả

* **Các khái niệm cơ bản:**
  + **Branch:** Các Branch (nhánh) đại diện cho các phiên bản cụ thể của một kho lưu trữ tách ra từ project chính. Cho phép theo dõi các thay đổi thử nghiệm thực hiện đối với kho lưu trữ và có thể hoàn nguyên về các phiên bản cũ hơn

Branch là cái dùng để phân nhánh và ghi lại luồng của lịch sử. Branch đã phân nhánh sẽ không ảnh hưởng đến branch khác nên có thể tiến hành nhiều thay đổi đồng thời trong cùng 1 repository.

Branch đã phân nhánh có thể chỉnh sửa tổng hợp lại thành 1 branch bằng việc hợp lại (merge) với branch khác

Các thành viên của nhóm sẽ tạo branch dùng riêng cho công việc của mình từ branch chính để không ảnh hưởng đến công việc của các thành viên khác. Sau đó, những thành viên đã hoàn thành công việc của mình sẽ thực hiện đưa thay đổi của mình vào branch chính



* **Fetch:** Lệnh git fetch tìm nạp các bản sao và tải xuống tất cả các tệp branch vào máy tính. Sử dụng để lưu các thay đổi mới nhất vào kho lưu trữ của bạn. Nó có thể tìm nạp nhiều branch cùng một lúc.
* **Fork:** Một fork là một bản sao của một kho lưu trữ (repository). Các lập trình viên thường tận dụng lợi ích của fork để thử nghiệm các thay đổi mà không ảnh hưởng đến dự án chín
* **Repository:** Kho lưu trữ Git chứa tất cả các tệp dự án (các file cũng như lịch sử các thay đổi được thực hiện đối với các file đó) bao gồm các branch, tags và commit
* **Staging area:** 
  + Với Git có 3 khu vực làm việc chính: Trên máy tính cá nhân, trên kho chứa mã nguồn(repository) và khu vực trung gian(staging area)

Staging Directory 
Working Directo 
(.index) 
add 
Commit 
Checkout 
Repository 

* Staging area là khu vực sẽ lưu trữ những thay đổi trên tập tin để nó có thể commit được, muốn commit được tệp tin phải nằm trong Staging area
* commit là một hành động để lưu lại một bản chụp của các sự thay đổi (xác nhận thay đổi) trong thư mục làm việc. Mỗi lần commit nó sẽ lưu lại lịch sử chỉnh sửa của mã nguồn kèm theo tên và địa chỉ email của người commit

=> muốn đưa tập tin lên repository thì phải commit nó trước rồi mới push mã nguồn lên

* **Index:** Git không có một thư mục Staging area chuyên dụng, thay vì điều này, nó sử dụng một file được gọi là chỉ mục(index). index là ảnh chụp của toàn bộ các file trong project chứ không chỉ là một danh sách các file đã bị thay đổi
* **Working Tree:** những thư mục được đặt trong sự quản lý của Git mà mọi người đang thực hiện công việc trong thực tế được gọi là working tree.

Giữa repository và working tree tồn tại một nơi gọi là index. Index là nơi để chuẩn bị cho việc commit lên repository

Index 
html 
Register 
css 

Khi đã thực hiện commit thì trạng thái sẽ không được ghi trực tiếp trong repository từ working tree, mà sẽ ghi trạng thái đã được thiết lập của index được xây dựng ở giữa đó. Vì thế, để ghi lại trạng thái của file bằng commit thì trước hết cần đăng ký file trong index

Với việc chèn index vào giữa như thế này, có thể thực hiện commit không bao gồm những file không cần thiết trong working tree, hay có thể đăng ký chỉ một phần thay đổi của file trong index rồi commit

* **Master** là nhánh chính của tất cả các repository của bạn. Nó nên bao gồm những thay đổi và commit gần đây nhất

develop 
Work in 
progress on 
"next release" 
master 
O 
Initial 
production 
version 
Next 
production 
release 
Next 
production 
release 

* **Merge:** kết hợp các pull requests để thêm các thay đổi từ nhánh này sang nhánh khác

* **Git vs Github:**
  + Git là tên gọi của một mô hình hệ thống, các máy tính có thể clone lại mã nguồn từ một repository ,

* GitHub là tên của một công ty cung cấp dịch vụ máy chủ repository công cộng, mỗi người có thể truy cập vào website trang chủ để tạo tài khoản trên đó và tạo ra kho chứa source của riêng mình khi làm việc
* Repository của Git được phân thành 2 loại là remote repository và local repository
  + Local Repository: là repository nằm trên chính máy tính của chúng ta, repository này sẽ đồng bộ hóa với remote repository bằng các lệnh của git.
  + Remote Repository: là repository được cài đặt trên server chuyên dụng. Ví dụ: GitHub, GitLab, Bitbucket,...

* **Các trạng thái trong Git Repository**

Unt racked 
Add the file 
Unmodif ied 
Edit the file 
Modified 
Staged 
Stage the file 
Remove the file 
Commit 

**Git Config**

Wednesday, July 13, 2022

3:38 PM

* /etc/gitconfig : Chứa giá trị cho tất cả người dùng và kho chứa trên hệ thống. Nếu sử dụng --system khi chạy git config, thao tác đọc và ghi sẽ được thực hiện trên tập tin này

* ~/.gitconfig : Riêng biệt cho tài khoản người dùng. Người dùng có thể chỉ định Git đọc và ghi trên tập tin này bằng cách sử dụng --global

Trên Window, file .gitconfig ở trong thư mục người dùng C:\Users\$user

* Tập tin config trong thư mục git (.git/config) của bất kỳ kho chứa nào đang sử dụng: Chỉ áp dụng riêng cho một kho chứa. Mỗi cấp sẽ ghi đè các giá trị của cấp trước nó, vì thế các giá trị trong .git/config sẽ được ưu tiên hơn các giá trị trong /etc/gitconfig

**Chia sẻ repo**

Wednesday, July 13, 2022

5:11 PM